

# Программирование Python

## Лекция 8

Кафедра ИВТ и ПМ  
ЗабГУ

2017

# План

Прошлые темы

Установка пакетов

Обзор пакетов

# Outline

Прошлые темы

Установка пакетов

Обзор пакетов

## Прошлые темы

- ▶ Как работает совместный цикл?
- ▶ Что будет выведено на экран?

```
for i in [22 -8, 7, 2]:  
    print(i/2, end=" ")
```

## Прошлые темы

- ▶ Как работает совместный цикл?
- ▶ Что будет выведено на экран?

```
for i in [22 -8, 7, 2]:  
    print(i/2, end=" ")
```

7.0 3.5 1.0

- ▶ Как работает функция Range?

# Прошлые темы

- ▶ Что такое множество?
- ▶ Какие существуют операции для работы со множествами?
- ▶ Когда использовать множества?

# Прошлые темы

- ▶ Что такое сортировка?
- ▶ Как работает алгоритм пузырьковой сортировки?
- ▶ Как работает алгоритм сортировки выбором?

# Прошлые темы

- ▶ Что такое модуль?
- ▶ Как подключить модуль из пакета?
- ▶ Как подключить конкретные функции из модуля?
- ▶ Как подключить всё содержимое модуля?



## Прошлые темы

- ▶ Какую работу можно выполнить с помощью matplotlib?
- ▶ Как построить простой график функции по набору точек?

# Прошлые темы

- ▶ Какую работу можно выполнить с помощью matplotlib?
- ▶ Как построить простой график функции по набору точек?

```
from matplotlib.pyplot import plot, show  
plot( [1,2,3,4,5], [11, 22, 33, 44, 55] )  
show()
```

- ▶ Как добавить координатную сетку?  
`grid(True)`

## Прошлые темы

- ▶ Как создать набор ординат по набору абсцисс?

## Прошлые темы

- ▶ Как создать набор ординат по набору абсцисс?

```
X = list (range(11) )
```

```
Y = [x**2 for x in X]
```

# Прошлые темы

- ▶ Как создать набор ординат по набору абсцисс?

```
X = list (range(11) )
```

```
Y = [x**2 for x in X]
```

- ▶ Как построить такой график?

```
plot( X, Y )
```

```
show()
```

## Прошлые темы

- ▶ Как сохранить график в файл?

```
savefig( "myplot.png" )
```

- ▶ Чем отличается график сохраненный в png файл от графика сохраненного в svg или pdf?
- ▶ Как построить гистограмму?

```
bar([1,2,3], [100, 300, 55])  
show()
```

## Прошлые темы

- ▶ Чем отличаются фактические параметры от формальных?
- ▶ Как использовать именованные параметры?

# Outline

Прошлые темы

Установка пакетов

Обзор пакетов



# Установка пакетов

- ▶ Большинство пакетов хранятся в сети — в репозитории <sup>1</sup>.
- ▶ Для установки пакетов используется отдельная утилита — `pip`.
- ▶ `Pip` самостоятельно скачивает и устанавливает указанный пакет.
- ▶ Для установки нужны права администратора (суперпользователя)
- ▶ `Pip` — утилита для командной строки.

---

<sup>1</sup>Репозиторий (репозиторий) - хранилище, в данном случае Python пакетов, поддерживаемое и обновляемое

## Где найти pip?

pip устанавливается вместе с Python.

- ▶ В ОС Linux эта программа доступна непосредственно в эмуляторе терминала:  
`user@PC:~$ pip help`
- ▶ В ОС Windows pip расположен в том же каталоге, что и python

Однако для работы с ним лучше использовать не командную строку windows, а отдельную - **Anaconda Prompt**.

# Установка пакетов

## Anaconda Prompt

Anaconda Prompt

```
(C:\ProgramData\Anaconda3) C:\Users\user>  
(C:\ProgramData\Anaconda3) C:\Users\user>  
(C:\ProgramData\Anaconda3) C:\Users\user>  
(C:\ProgramData\Anaconda3) C:\Users\user>pip
```

Usage:

pip <command> [options]

Commands:

install

Install packages.

download

Download packages.

uninstall

Uninstall packages.

freeze

Output installed packages in re

list

List installed packages.

show

Show information about installe

check

Verify installed packages have

cies.

search

Search PyPI for packages.

wheel

Build wheels from your requirem

hash

Compute hashes of package archi

completion

A helper command used for comma

help

Show help for commands.

General Options:

## Установка пакетов

Если же Anaconda Prompt не установлена, то используется командная строка Windows.

Для обращения к `pip` нужно указать полный путь до `pip.exe` или перейти в каталог программы, а затем запустить её непосредственно оттуда:

```
./pip.exe help
```

Стоит помнить, что для установки пакетов потребуются права администратора (суперпользователя) Далее будет приведены

примеры работы с `pip` в Linux, однако принципы работы с программой одинаковы для всех ОС.

# Установка пакетов

Выбрать Windows PowerShell

Windows PowerShell

(C) Корпорация Майкрософт, 2009. Все права защищены.

PS C:\Users\user>

PS C:\Users\user>

PS C:\Users\user> cd C:\ProgramData\Anaconda3\Scripts

PS C:\ProgramData\Anaconda3\Scripts>

PS C:\ProgramData\Anaconda3\Scripts>

PS C:\ProgramData\Anaconda3\Scripts> .\pip.exe help

Usage:

pip <command> [options]

Commands:

install

download

uninstall

freeze

list

show

check

search

wheel

hash

completion

help

Install packages.

Download packages.

Uninstall packages.

Output installed packages in re

List installed packages.

Show information about installe

Verify installed packages have

Search PyPI for packages.

Build wheels from your requirem

Compute hashes of package archi

A helper command used for comma

Show help for commands.

General Options:

-h, --help

--isolated

-v, --verbose

Show help.

Run pip in an isolated mode, ig

environment variables and user

Give more output. Option is add

# Использование pip

pip `команда` [`опции`]

## Команды pip

- ▶ `install <имя пакета>` — установка
- ▶ `uninstall <имя пакета>` - удаление
- ▶ `show <имя пакета>` - показывает информацию о пакете
- ▶ `search <имя пакета>` поиск пакетов в репозитории.
  
- ▶ `help` - короткая справка

## Установка пакета

```
pip install numpy
```

Вывод программы

```
sudo pip install numpy
```

Collecting numpy

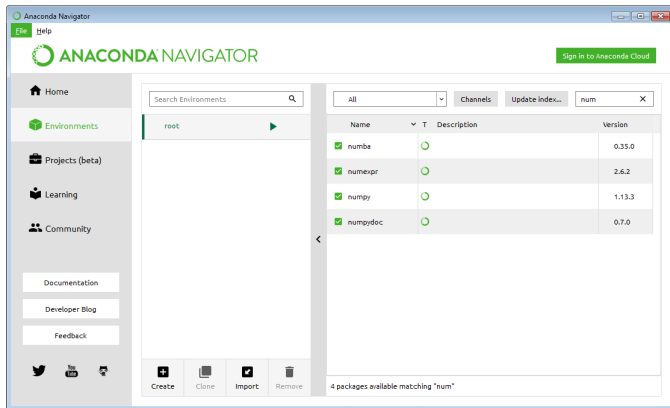
Downloading numpy-1.13.3-cp27-cp27mu-manylinux1\_x86\_64.whl

100% || 16.7MB 49kB/s

Installing collected packages: numpy

Successfully installed numpy-1.13.3

# установка пакетов в Anaconda





# Outline

Прошлые темы

Установка пакетов

Обзор пакетов

# Некоторые пакеты

- ▶ **OS** - работа с операционной системой
- ▶ **sys** обеспечивает доступ к некоторым переменным и функциям, взаимодействующим с интерпретатором python.
- ▶ **time** - работа со временем
- ▶ **datetime** - работа с датами (и временем)
- ▶ **re** - обработка регулярных выражений
- ▶ **requests** - http запросы, получение данных из Интернета
- ▶ **argparse, getopt** - работа с параметрами запуска скрипта

## Некоторые пакеты

- ▶ **PIL** - работа с растровой графикой
- ▶ **pygame** - фреймворк создания простых игр
- ▶ **tkinter** - фреймворк для создания приложений с графическим интерфейсом пользователя
- ▶ **simple-crypt** - предельно простое шифрование (использует AES256)
- ▶ **Flask** - микрофреймворк для создания сайтов
- ▶ **Selenium, Mechanize** - эмуляция браузера
- ▶ **Scapy** - обработка сетевых пакетов

# Популярные математические пакеты

- ▶ **numpy** - работа с матрицами и многомерными массивами
- ▶ **scipy** - научные и инженерные вычисления, использует numpy;
- ▶ **matplotlib** - построение графиков и диаграмм;
- ▶ **seaborn** - визуализация статистических данных, эстетичнее чем matplotlib;
- ▶ **mpld3** - использование D3.js для построения интерактивных matplotlib графиков в окне браузера;
- ▶ **pandas** - анализ данных: статистики, регрессия, визуализация и т.п.

# Ссылки и литература

Ссылка на слайды

[github.com/VetrovSV/Programming](https://github.com/VetrovSV/Programming)